



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/992,099	11/05/2001	Daniel Scott Jorgenson	10011598 -1	7507

7590 06/17/2005

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER

COFFY, EMMANUEL

ART UNIT	PAPER NUMBER
----------	--------------

2157

DATE MAILED: 06/17/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/992,099

Applicant(s)

JORGENSEN, DANIEL SCOTT

Examiner

Emmanuel Coffy

Art Unit

2157

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on ____.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-25 is/are pending in the application.
- 4a) Of the above claim(s) ____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) ____ is/are allowed.
- 6) ☐ Claim(s) 1-25 is/are rejected.
- 7) ☐ Claim(s) ____ is/are objected to.
- 8) ☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on ____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. ____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date ____.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date ____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: ____.

Response to Amendment

1. This action is responsive to the Amendment filed on March 28, 2005. Claims 1-25 are pending. Claims 1-2, 5, 12-14, 22, 25 are amended. Claims 1-25 represent a "System & Method for Maintaining Consistent Independent Server-side State Among Collaborating Servers."

Response to Arguments

2. Applicant's narrative arguments contributed to a better understanding of the invention and as such made possible a better articulation of what the application lacks to meet the novel requirements over prior art and common knowledge in the art. These arguments have been fully considered and albeit eloquent they are not persuasive.

2.1 On page 11 of the remarks applicant states: " In contrast, claimed embodiments of the present invention are directed to the maintenance, when appropriate, of server-side state of interest to a single user in systems having multiple, collaborating servers. Thus, claimed embodiments are directed to the proper maintenance of single-user state information, such as user session state, and even then only when appropriate. The Office Action doesn't seem to appreciate this significant distinction. The claim language, which embodies this distinction, will be discussed in more detail below."

Rather than discussing the language, applicant is advised to incorporate the language within the claim to the extent possible.

2.2 On page 13 of the remarks, applicant states: "After the second server/cluster notices these new events, it repeats them locally and automatically). That is, the second server/cluster may do whatever the event entails (it depends on the event

and the architecture of the system in regard to the event and the state affected by the event) and does not necessarily need to conduct the first server/cluster (or any other server/cluster) to get fresh data or server-side state information, unlike embodiments of the '390 patent. This alternative embodiment enjoys an advantage in that it further abstracts the state change so that servers (which could be of quite different architectures) can have their actual state representations completely decoupled from one another (see discussion in specification pp. 29-35)."

It is not understood how a state change is abstracted. It is appreciated that applicant attempted to distinguish the application at bar with '390 patent. However, an event inherently engenders a state change, i.e., when an event is performed such as login or logout there is a state change inherently associated with such event. Whether the change is acknowledged or acted upon is irrelevant to the existence of the actual change; the fact is a change is always associated with an event in this context. Replication of the change or the absence thereof is not a patentable distinction over the prior art or common knowledge in the art.

The Examiner maintains the arguments presented in the First Office Action as outlined below and the rejection is therefore sustained. The dependent claims stand rejected as articulated in the First Office Action and all objections not addressed in Applicant's response are herein reiterated.

Claim Rejections - 35 USC § 102

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

3. Claim 1 is rejected under 35 U.S.C. §102(e) as being anticipated by Prasad et al. (US 6,539,381.)

Prasad teaches the invention as claimed including a system and method for synchronizing database information over a communications network distributed among a plurality of servers. The system allows each server to track the state of a replica on each of the servers, and changes to the servers are communicated between the servers along with their states. (See abstract).

Claim 1:

As to claim 1, Prasad teaches a method of replicating state information among a plurality of collaborating servers connected to a network, the method comprising: (See Fig. 1)

determining at a subscriber server from information stored on a client computer whether an event has occurred on a publisher server which event implicates a need for state change on the subscriber server; and (See col. 3, lines 41-54.)

if such an event has occurred, replicating state effects of the event into state on the subscriber server. (See col. 3, lines 41-56.)

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

4. Claims 2-25 are rejected under 35 U.S.C. §103(a) as being unpatentable over Prasad et al. (US 6,539,381) in view of Sedlar (US 6,549,916.)

Claim 2:

As to claim 2, Prasad teaches the method of claim 1, wherein the step of determining at a subscriber server from information stored on a client computer whether an event has occurred on a publisher server further comprises:

receiving an event queue stored on the client computer to the subscriber server; determining whether any events recorded in the event queue are not yet replicated on the subscriber server; and determining whether any such events require replication on the subscriber server. (See col. 3, lines 44-51. - Prasad teaches determining whether any such events require replication on the subscriber server by comparing the timestamp of the replica.)

Prasad does not explicitly teach event queue. However, Sedlar extensively discloses event queue mechanism at col. 28, lines 21-50.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the data base synchronization system as taught by Prasad with the event notification system disclosed by Sedlar. This system would be

desirable in that the file system event notification mechanism allows a file cache to be proactively updated so that it always reflects the current state of the files at their original locations.

Claim 3:

As to claim 3, Prasad teaches the method of claim 2, wherein the step of determining whether any events in the event queue are not yet replicated on the subscriber server further comprises comparing the event queue retrieved from the client computer with a most-recently-experienced event queue recorded by the subscriber server.

Prasad teaches determining whether any such events require replication on the subscriber server by comparing the timestamp of the replica. Prasad does not explicitly teach event notification. However, Sedlar extensively discloses event queue mechanism based on proactive notification upon occurrence at col. 28, lines 21-50.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the data base synchronization system as taught by Prasad with the event notification system disclosed by Sedlar. This system would be desirable in that the file system event notification mechanism allows a file cache to be proactively updated so that it always reflects the current state of the files at their original locations. Additionally, no overhead of repeated polling would be incurred comparing the event queue retrieved from the client computer with a most-recently-experienced event queue recorded by the subscriber server.

Claim 4:

As to claim 4, Prasad teaches the method of claim 3, wherein the event queue retrieved from the client computer is recorded as the most-recently-experienced event queue by the subscriber server following the comparison.

Prasad does not explicitly teach event notification. However, Sedlar extensively discloses event queue mechanism based on proactive notification upon occurrence at col. 28, lines 21-50.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the data base synchronization system as taught by Prasad with the event notification system disclosed by Sedlar. This system would be desirable in that no overhead of repeated polling would be incurred in determining the most-recently-experienced event queue by the subscriber server following the comparison to be recorded.

Claim 5:

As to claim 5, Prasad teaches the method of claim 2, wherein the step of determining whether any unreplicated events in the event queue require replication on the subscriber server further comprises: determining what state on the subscriber server is relevant to the subscriber server at the time; and determining if replicating such an event would effect change to such state. (See col. 3, lines 14-20. - Prasad teaches transmitting state information of a first server to a third server. Prasad further teaches a mechanism to determine if replicating such an event would effect change to such state. (See col. 14, lines 13-19.))

Prasad does not explicitly teach file event queue. However, Sedlar extensively discloses file event queue at col. 28, lines 30-51.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the data base synchronization system as taught by Prasad with the file event queue system disclosed by Sedlar. This system would be desirable in that the file event queue mechanism allows a file to change state.

Claim 6:

As to claim 6, Prasad teaches the method of claim 2, wherein the step of determining whether any unreplicated events in the event queue require replication on the subscriber server further comprises determining if undertaking replication of one such event can be skipped by the subscriber server due to the existence of another such event whose replication by the subscriber server would suffice for the first. (See col. 4, lines 1-14. - Prasad teaches transmitting a change to a server due to existence of a timestamp of the replica of the second server.)

Prasad does not explicitly teach event replication. However, Sedlar extensively discloses event notification at col. 27, lines 1-10.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the data base synchronization system as taught by Prasad with the event notification system disclosed by Sedlar. This system would be desirable in that the file system event notification mechanism allows a file cache to be proactively updated so that it always reflects the current state of the files at their original locations.

Claim 7:

As to claim 7, Prasad teaches the method of claim 1, wherein the step of replicating the state effects of an event into state on the subscriber server further comprises copying state data from the event queue into subscriber server state. (See col. 3, lines 25-40. - Prasad teaches replicating state information of a first server to a third server.).

Prasad does not explicitly teach file event queue. However, Sedlar extensively discloses file event queue at col. 27, lines 11-24.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the state replication system as taught by Prasad with the file event queue system disclosed by Sedlar. This system would be desirable in that the file event queue mechanism allows a file to change state.

Claim 8:

As to claim 8, Prasad teaches the method of claim 1, wherein the step of replicating the state effects of an event into state on the subscriber server further comprises copying state data from another server computer into subscriber server state. (See col. 3, lines 60-65. - Prasad teaches the change is initiated by an update received from another server.).

Prasad does not explicitly teach file event queue. However, Sedlar extensively discloses file event queue at col. 27, lines 11-24.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the state replication system as taught by Prasad with

the file event queue system disclosed by Sedlar. This system would be desirable in that the file event queue mechanism allows a file to change state.

Claim 9:

As to claim 9, Prasad teaches the method of claim 8, wherein the state data is copied by the subscriber server over the network from the publisher server.

Prasad teaches replicating state information of a first server to a third server.
(See col. 3, lines 25-40; See also col. 5, lines 1-30.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to use the state replication system as taught by Prasad.

Claim 10:

As to claim 10, Prasad teaches the method of claim 8, wherein the state data is copied by the subscriber server over the network from a database server.

Prasad teaches replicating state information of a first server to a third server.
(See col. 3, lines 25-40.) Furthermore, Prasad teaches a change in the first data base and the state of the first database to an intermediate entity having a second copy of the database.

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to use the state replication system as taught by Prasad.

Claim 11:

As to claim 11, Prasad teaches the method of claim 8, wherein the state data copied by the subscriber server over the network from another server is converted from a first data format into a second data format.

Prasad does not explicitly teach data format conversion. However, Sedlar specifically discloses file transmission using different protocols at col. 29, lines 1-10.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the state replication system as taught by Prasad with the file transmission system disclosed by Sedlar. This system would be desirable in that the file transmission mechanism allows for transmission to different machines with disparate protocols.

Claim 12:

As to claim 12, Prasad teaches the method of claim 1, further comprising: processing an event on a publisher server which causes state change on the publisher server; retrieving an event queue stored on a client computer to the publisher server; adding a new event descriptor characterizing the event to the event queue; and sending the event queue from the publisher server to be stored on the client computer.

(See col. 3, lines 25-40; See also col. 5, lines 1-30. - Prasad teaches replicating state information of a first server to a third server.)

Prasad does not explicitly teach file event queue. However, Sedlar extensively discloses file event queue at col. 27, lines 11-24 and lines 55-61; col. 29, lines 10-15.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the state replication system as taught by Prasad with the file event queue system disclosed by Sedlar. This system would be desirable in that the file system event notification mechanism allows a file cache to be proactively updated so that it always reflects the current state of the files at their original locations.

Claim 13:

As to claim 13, Prasad teaches the method of claim 2, wherein the event queue is stored on the client computer within a transport mechanism of a cookie. (See col. 3, lines 54-56. - A cookie is a block of data that a server returns to a client in response to a request from the client. (μSoft Computer Dictionary, 5th ed., p. 129. Prasad teaches storing a new time stamp (a block of data) at the second server.)

Prasad does not explicitly teach file event queue. However, Sedlar extensively discloses file event queue at col. 27, lines 11-24 and lines 55-61; col. 29, lines 10-15.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the state replication system as taught by Prasad with the file event queue system disclosed by Sedlar. This system would be desirable in that the file system event notification mechanism allows a file cache to be proactively updated so that it always reflects the current state of the files at their original locations.

Claim 14:

As to claim 14, Prasad teaches the method of claim 2, wherein the event queue includes: one or more event descriptors;

a uniqueness provision; and (See col. 11, lines 14-16. - Prasad teaches the notion of a unique identifier.)

a specified duration. (col. 14, lines 46-49. - , Prasad teaches a preset interval of time)

Prasad does not explicitly teach file event descriptors. However, Sedlar extensively discloses file event descriptors at col. 27, lines 55-61.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the data base synchronization system as taught by Prasad with the event descriptors disclosed by Sedlar. This system would be desirable in that notification rules may then be created.

Claim 15:

As to claim 15, Prasad teaches the method of claim 12, wherein each event descriptor includes: a characterization of the general type of event;

zero or more arguments characterizing the specific instance of the event;

a discernible ordinal position within an event queue. (See col. 11, line 59-col. 12, line 17. - Prasad teaches the notion of a discernible position.)

Prasad does not explicitly teach file event descriptors or specific instance of events. However, Sedlar extensively discloses file event descriptors at col. 27, lines 55-61), characterization at lines 62-66. Furthermore, Sedlar discloses event queue mechanism based on proactive notification upon occurrence at col. 27, lines 1-10.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the notion of discernible ordinal position system as taught by Prasad with the event notification system disclosed by Sedlar. This system would be desirable in that the file system event notification mechanism allows a file cache to be proactively updated so that it always reflects the current state of the files at their original locations. Additionally, no overhead of repeated polling would be incurred comparing the event queue retrieved from the client computer with a most-recently-experienced event queue recorded by the subscriber server.

Claim 16:

As to claim 16, Prasad teaches the method of claim 14, wherein the uniqueness provision is a unique value in the form of a timestamp. (See col. 13, lines 58-66; col. 11, lines 14-16, line 63- col. 12, line 5.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to use the unique identifier in the form of a time stamp as taught by Prasad.

Claim 17:

As to claim 17, Prasad teaches the method of claim 14, wherein the uniqueness provision is a unique value in the form of a pseudorandom datum. (See col. 13, lines 58-63.) Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to use the unique identifier in the form of a pseudorandom datum as taught by Prasad.

Claim 18:

As to claim 18, Prasad teaches the method of claim 12, wherein the step of retrieving an event queue stored on a client computer to a publisher server further comprises allocating an initial event queue if no event queue is yielded by the retrieval.

Prasad does not explicitly teach event notification. However, Sedlar extensively discloses event queue mechanism based on proactive notification upon occurrence at col. 27, lines 1-10.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the data base synchronization system as taught by Prasad with the event notification system disclosed by Sedlar. This system would be

desirable in that no overhead of repeated polling would be incurred in determining the most-recently-experienced event queue by the subscriber server following the comparison to be recorded.

Claim 19:

As to claim 19, Prasad teaches the method of claim 12, wherein the step of adding a new event descriptor characterizing an event to an event queue further comprises adding the event descriptor such that it is ordinally maximal within the event queue. (See col. 11, line 59-col. 12, line 17. - Prasad teaches the notion of a discernible subordinal position.)

Prasad does not explicitly teach file event descriptors. However, Sedlar extensively discloses file event descriptors at col. 27, lines 55-61), event queue at col. 28, lines 31-39. Furthermore, Sedlar discloses event queue mechanism based on proactive notification upon occurrence at col. 27, lines 1-10.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the notion of discernible ordinal position system as taught by Prasad with the event notification system disclosed by Sedlar. This system would be desirable in that the file system event notification mechanism allows a file cache to be proactively updated so that it always reflects the current state of the files at their original locations. Additionally, no overhead of repeated polling would be incurred comparing the event queue retrieved from the client computer with a most-recently-experienced event queue recorded by the subscriber server.

Claim 20:

As to claim 20, Prasad teaches The method of claim 12, wherein the step of adding a new event descriptor characterizing an event to an event queue further comprises removing those event descriptors from the event queue which are no longer needed by any collaborating server. (See col. 14, line 60-col. 15, line 5. - Prasad teaches the notion of merging state vectors.)

Prasad does not explicitly teach file event descriptors. However, Sedlar extensively discloses file event descriptors at col. 27, lines 55-61, event queue at col. 28, lines 31-39. Furthermore, Sedlar discloses deletion of oldest non-tagged version of a file at col. 34, lines 24-28.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the notion of state merger as taught by Prasad with the event notification system disclosed by Sedlar. This system would be desirable in that the system would be more efficient.

Claim 21:

As to claim 21, Prasad teaches the method of claim 12, wherein the event queue sent to the client computer is recorded as the most-recently-experienced event queue by the publisher server.

Prasad does not explicitly teach event notification. However, Sedlar extensively discloses event queue mechanism based on proactive notification upon occurrence at col. 28, lines 21-50.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the data base synchronization system as taught by

Prasad with the event notification system disclosed by Sedlar. This system would be desirable in that no overhead of repeated polling would be incurred in determining the most-recently-experienced event queue by the subscriber server following the comparison to be recorded.

Claim 22:

As to claim 22, Prasad teaches a system for replicating state information among a plurality of collaborating servers connected to a network, the system comprising: (See Fig. 1)

logic configured to determine at a subscriber server from information stored on a client computer whether an event has occurred on a publisher server which event implicates a need for state change on the subscriber server; and (See col. 3, lines 41-54; col. 19 line 64-col 20, line 33; col. 22, lines 9-55.)

logic configured to replicate state effects of the event into state on the subscriber server, if such an event has occurred. (See col. 3, lines 41-56; col. 19 line 64-col 20, line 33.)

Prasad does not explicitly teach event notification. However, Sedlar extensively discloses event notification at col. 27, lines 1-10.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the data base synchronization system as taught by Prasad with the event notification system disclosed by Sedlar. This system would be desirable in that the file system event notification mechanism allows a file cache to be proactively updated so that it always reflects the current state of the files at their original locations.

Art Unit: 2157

Claim 23:

As to claim 23, Prasad teaches the system of claim 22, wherein the logic configured to determine further comprises:

logic configured to retrieve an event queue stored on the client computer to the subscriber server; (See col. 19, line 64-col. 20, line 20.)

logic configured to determine whether any events recorded in the event queue are not yet replicated on the subscriber server; and (See col. 19, line 64-col. 20, line 20.)

logic configured to determine whether any such events require replication on the subscriber server. (See col. 19, line 64-col. 20, line 20. - Prasad teaches determining whether any such events require replication on the subscriber server by comparing the timestamp of the replica (See col. 3, lines 44-51.))

Prasad does not explicitly teach event notification. However, Sedlar extensively discloses event queue mechanism at col. 28, lines 21-50.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the data base synchronization system as taught by Prasad with the event notification system disclosed by Sedlar. This system would be desirable in that the file system event notification mechanism allows a file cache to be proactively updated so that it always reflects the current state of the files at their original locations.

Claim 24:

As to claim 24, Prasad teaches the system of claim 23, wherein the logic configured to determine whether any unreplicated events in the event queue require replication on the subscriber server further comprises:

logic configured to determine what state on the subscriber server is relevant to the subscriber server at the time; and (See col. 3, lines 14-20. - Prasad teaches transmitting state information of a first server to a third server.)

logic configured to determine if replicating such an event would effect change to such state. See col. 14, lines 13-19; col. 18, lines 9-55. - Prasad further teaches a mechanism to determine if replicating such an event would effect change to such state.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to use the evaluation of state replica and synchronization system as taught by Prasad.

Claim 25:

As to claim 25, Prasad teaches the system of claim 22, further comprising:

logic configured to process an event on a publisher server which causes state change on the publisher server; (See col. 3, lines 25-40; See also col. 5, lines 1-30; col. 18, lines 9-55. - Prasad teaches replicating state information of a first server to a third server.)

logic configured to retrieve an event queue stored on a client computer to the publisher server;

logic configured to add a new event descriptor characterizing the event to the event queue; and

logic configured to send the event queue from the publisher server to be stored on the client computer.

Prasad does not explicitly teach file event queue. However, Sedlar extensively discloses file event queue at col. 27, lines 11-24 and lines 55-61; col. 29, lines 10-15.)

Hence, it would have been obvious at the time of the invention for an artisan of ordinary skill in the art to combine the state replication system as taught by Prasad with the file event queue system disclosed by Sedlar. This system would be desirable in that the file system event notification mechanism allows a file cache to be proactively updated so that it always reflects the current state of the files at their original locations.

5. THIS ACTION IS MADE FINAL.

Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.


Art Unit: 2157

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Emmanuel Coffy whose telephone number is (703) 305-0325. The examiner can normally be reached on 8:30 - 5:00 P.M.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ario Etienne can be reached on (703) 308-7562. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Emmanuel Coffy
Patent Examiner
Art Unit 2157

***EC
June 10, 2005



SALEH NAJJAR
PRIMARY EXAMINER